Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-46610

A COMPARATIVE STUDY ON ACCURACY AND EFFICIENCY OF METAMODELS FOR LARGE INDUSTRIAL DATASETS

Arun K. Subramaniyan
GE Global Research, 1 Research Circle
Niskayuna, NY, USA.

Liping Wang
GE Global Research, 1 Research Circle
Niskayuna, NY, USA.

Don Beeson
GE Aviation, 1 Neumann Way
Cincinnati, OH, USA.

John Nelson
GE Aviation, 1 Neumann Way
Cincinnati, OH, USA.

Richard Berg
GE Aviation, 1 Neumann Way
Cincinnati, OH, USA.

Randall Cepress
GE Aviation, 1 Neumann Way
Cincinnati, OH, USA.

ABSTRACT

This paper provides a comparative study on accuracy and efficiency of metamodels constructed from large datasets. Two examples inspired by large industrial applications are used to identify the best metamodeling technique. Artificial Neural Networks, Radial Basis Functions, Gaussian Process and Non-linear regression are used to build metamodels. The examples used showcase a broad range of industrial applications in aircraft engines and gas turbines. Although Radial Basis Functions and Gaussian Process models are robust for small data sets, their high computational cost for large datasets reduces their practical application. ANN models are found to perform optimally when large number of training points are readily available and the accuracy requirements are high.

INTRODUCTION

Simplified approximate models of data generated by experiments and high-fidelity simulation models are called metamodels. Metamodels are now widely accepted and used in many industries for complex engineering system design studies where many design-evaluate-redesign cycles are required [1, 2]. When accurately built, metamodels can be used to replace expensive simulation models to enable practical and affordable optimization and probabilistic design studies [3, 4]. Most of the existing studies describe replacing slow running simulations as the primary use of metamodels [5, 6, 7]. Thus, the comparative studies on accuracy and efficiency of the metamodeling techniques have been performed based on a limited number of training data generated from expensive simulations [8]. The construction and usefulness of metamodels built using large datasets has not been studied in detail. However, there are many engineering applications where thousands (or more) of data points are readily available. These situations include field test data such as engine flight data, existing data from past simulation runs, etc. If processed correctly, these datasets can be used to build very useful metamodels for fast executing prediction of the relationships between engineering system inputs and outputs. Some typical applications include prognostics and health monitoring (PHM), onboard control systems requiring high accuracy and fidelity, etc.

This paper provides a comparative study on accuracy and efficiency of metamodels constructed from large datasets.

"What is the best metamodel to use?" is a highly subjective question that depends on many factors such as the number of design parameters, the size of training data set, presence or absence of noise and outliers, required accuracy for predictions, speed of execution, etc. For example, some applications require very high accuracy (~0.1% error) for predictions but the data set may not have any noise or outliers. Other applications have very noisy data, however, the metamodels are required to predict only the trends and shifts accurately rather than actual values. The above examples are at two opposite ends of the requirements spectrum. Clearly, the same metamodeling technique may not be able to satisfy both requirements. The goal of this study is to identify the best metamodeling technique for the former case, i.e., large dataset with high accuracy and smoothness requirements. The metamodeling techniques to be compared include Artificial Neural Networks (ANN) [9, 10, 11], Radial Basis Functions (RBF) [12, 13], Gaussian Process (GP) [14, 15] and Nonlinear Regression (N-LR) [16, 17]. A brief introduction of some of the above metamodeling methods is given below.

Gaussian Process (GP) Metamodels

Gaussian process is a non-parametric metamodeling technique. GP models are built by first calculating a stationary or non-stationary covariance matrix of the training data. The stationary and non-stationary covariance functions are listed in Equations (1) and (2).

$$C_{ij} = \theta_1 \exp\left\{-\frac{1}{2}\sum_{l=1}^{L}\frac{\left(x_i^{(l)} - x_j^{(l)}\right)^2}{r_l^2}\right\} + \theta_2 \qquad (1)$$

$$C_{ij} = \theta_1 \exp\left\{-\frac{1}{2}\sum_{l=1}^{L}\frac{\left(x_i^{(l)} - x_j^{(l)}\right)^2}{r_l^2}\right\} + \theta_2 + \delta_{ij}\theta_3 \qquad (2)$$

where θ_1 is the vertical scale amplitude, θ_2 is the vertical offset (mean), r_l is the "radius of influence" of each x_l, L is the number of input variables and θ_3 is an independent noise parameter.

The hyper parameters of the GP model are calculated by minimizing the following objective function:

$$Objective = -\tfrac{1}{2} y_N^T C_N^{-1} y_N - \tfrac{1}{2}\ln|C_N| + \ln P(\theta) \qquad (3)$$

where y_N is the vector of the output variable Y from the training data set, $P(\theta)$ is the prior distributions of the optimized parameters, C_N is the covariance matrix. Since the objective function is generally complex with many local minima, global optimization techniques such as genetic algorithms are used to solve the optimization problem.

Based on the covariance matrix and the inverse of the covariance, the Gaussian process computes the predictions as listed below.

$$y_{N+1} = k^T C_N^{-1} y_N \qquad (4)$$

where

$$C_{N+1} = \begin{bmatrix} \begin{bmatrix} & C_N & \end{bmatrix} & \begin{bmatrix} k \end{bmatrix} \\ \begin{bmatrix} & k^T & \end{bmatrix} & \begin{bmatrix} \alpha \end{bmatrix} \end{bmatrix}$$

$$k^T = \begin{bmatrix} C_{1,N+1} & C_{2,N+1} & \cdots & C_{N,N+1} \end{bmatrix}$$

Radial Basis Function Metamodels

Radial Basis Functions (RBF) are non-parametric models similar to GP models [18, 19]. This method uses linear combinations of radially symmetric functions based on Euclidean distance or other such metric to approximate response functions. A simple RBF kernel can be written as:

$$Y(x) = \sum_{i=1}^{N} \lambda_i \varphi(\|x - x_i\|_2) \qquad (5)$$

where λ_i are interpolation coefficients to be determined, N is the number of sample or training data points. The radial basis functions φ are functions of the Euclidean norm $\|x - x_i\|_2$ from node i, which is the radial distance ***r*** of the point *x* from the center x_i.

The unknown interpolation coefficients λ_i can be determined by minimizing the norm *J* [20,21] given in

$$J = [f(x_k) - \sum_{i=1}^{N} \lambda_i \varphi(\|x_k - x_i\|_2)]^2 \qquad (6)$$

$$k = 1,2,3,\ldots,N$$

Artificial Neural Network (ANN) Metamodels

ANNs used for building meta-models that fit data consist of two layers: a sigmoid layer followed by a linear output layer. It is known that neural networks with a sigmoid layer along with a linear output layer can be trained to approximate functions with finite number of discontinuities [22]. The process of training a multi-layer network by changing its weights and biases is called backpropagation [23]. The hidden sigmoid layer consists of p neurons and the input vector {x} has m elements. The weight matrix $W1$ of the hidden layer is of size $p \times m$. In general, the layer names are given based on the functions used in the layer.

In the case of a multi-layer network, the output of a preceding layer becomes the input for the next layer. Thus, the output from a two layer network is calculated in two steps as listed below.

$$\begin{aligned} Hidden\,Layer: \{y'\}_{p\times 1} &= f\left([W]_{p\times m}\{x\}_{m\times 1} + \{b\}_{p\times 1}\right) \\ Output\,Layer: \; y_{1\times 1} &= Lf\left([OW]_{1\times p}\{y'\}_{p\times 1} + ob_{1\times 1}\right) \end{aligned} \quad (7)$$

where
W is the weight matrix of the hidden layer
x is the input vector
f is the function in the hidden layer
b is the bias vector in the hidden layer
$\{y'\}$ is the output of the hidden layer
Lf is the output function
OW is the output layer weight vector
Ob is the output layer bias
y is the ANN prediction (output)

Large data sets inspired from industrial applications with varying degrees of nonlinearity are used to assess the modeling techniques. Although there is no "silver bullet" that satisfies all requirements, a clear winner emerges for particular set of requirements and available datasets.

It should be noted that the examples are only inspired by industrial applications and bear no resemblance to actual datasets used in GE aircraft engine or gas turbine applications.

APPLICATION-A

This first application is an example dataset that reflects the size and non-linearity of many datasets used in control systems. The data contains three independent variables and one dependent variable. The scaled dataset is shown in Figure 1. Note that the response is plotted on the vertical axis with the third independent parameter (X3) shown as colors. This rather unusual plot was chosen to highlight the variation in both the independent and dependent variables.

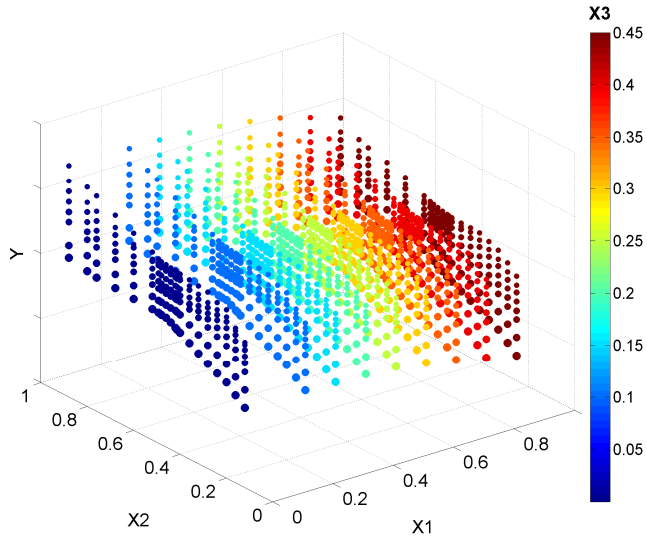

Figure 1 Non-linear dataset with three independent parameters (Y axis values removed intentionally).

The above dataset is medium sized with ~1400 points. The requirements for the metamodels are the following:

- Maximum absolute prediction error 0.5 %
- Predictions between grid locations and off-grid locations should be smooth

Although the nonlinearity in the dataset is uniform for most of the data range, the strict accuracy requirements over the entire range of independent parameters coupled with the smoothness requirement make this problem challenging.

The following metamodeling techniques were used to compare their performance in modeling the above dataset: nonlinear regression, Radial Basis Functions (RBF), Gaussian Process (GP) and Artificial Neural Networks (ANN).

All the available points (1441) were used to build the nonlinear regression model. The RBF and GP models were built using 100 points extracted from the original dataset using points that are closest to an optimized Latin-hypercube (OLH) design with the same range for the independent parameters. Fewer points were used for GP and RBF models because the computational cost for building GP and RBF models with greater than 300 points is prohibitive. Since there are three independent parameters, ideally training points greater than 30 should suffice to produce accurate RBF and GP models. Ten points were used to cross-validate the RBF model. The RBF constant was optimized to reduce the cross-validation error. A genetic algorithms (GA) based optimizer was used to optimize the hyperparameters for generating an interpolant GP model.

Two ANN models were built: one with the same training points as the GP and RBF models and another ANN model was built using 70 % (1008) of the dataset as training points and 15 % (216) each for validation and testing. A two layer ANN with hidden sigmoid layer combined with a linear output layer was used. Forty nodes were used in the sigmoid layer. The Levenberg-Marquardt [24, 25, 26] algorithm was used to optimize the weight matrices and bias vectors of the ANN nodes with an objective to minimize the root-mean-square (RMS) error of predicted values.

Metamodel Accuracy

The quality metrics for model predictions of the above models are listed in Table 1. The mean (L1 norm), root mean-squared error (RMSE) and maximum absolute errors were calculated from predictions at all available data points using the following formulae.

$$Mean\ \ error = \frac{\sum_{i=1}^{n} \left| Y_i^{pred} - Y_i^{actual} \right|}{n}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(Y_i^{pred} - Y_i^{actual} \right)^2}{n}} \qquad (4)$$

$$Max\ \%\ error = 100 \times \max \left| \frac{Y_i^{pred} - Y_i^{actual}}{Y_i^{actual}} \right| ; \ i = 1,2...n$$

where Y_i^{pred} are the predicted values

Y_i^{actual} are the actual values

n is the total number of data points

Table 1 Prediction errors of various metamodels.

Model	Mean Error	RMSE	Max. Abs. Error (%)
ANN (100 pts)	5.92	8.65	1.83
RBF (100 pts)	5.53	8.78	3.31
GP (100 pts)	6.65	12.33	4.77
N-LR (1440 pts)	10.76	13.67	2.95
ANN (1000 pts)	1.23	1.62	0.48

The metrics in Table 1 alone are insufficient to assess whether the metamodels have satisfied the requirements stated above. Figures 2-6 show regions in the dataset with errors greater than 0.25 % (maximum allowable error is 0.5 %).

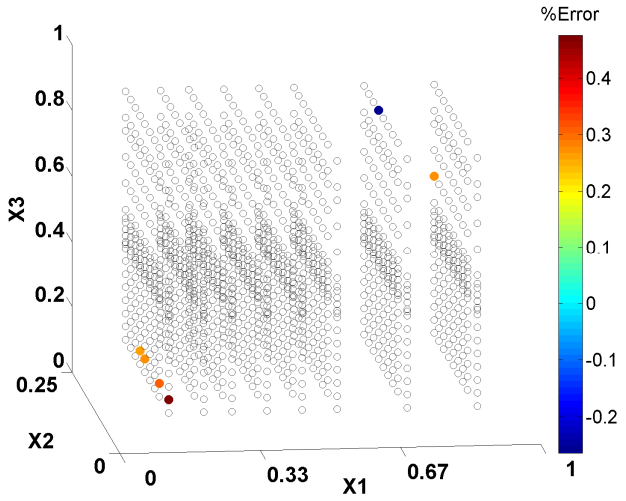

Figure 2 Points with ANN prediction error > 0.25 % (1000 training points).

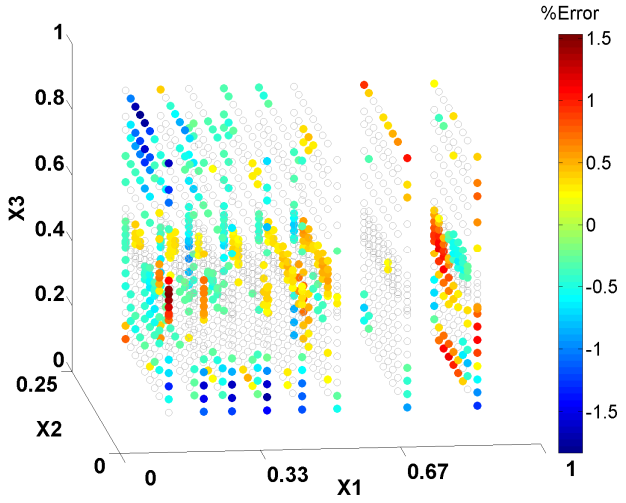

Figure 3 Regions with ANN prediction error > 0.25 % (100 training points)

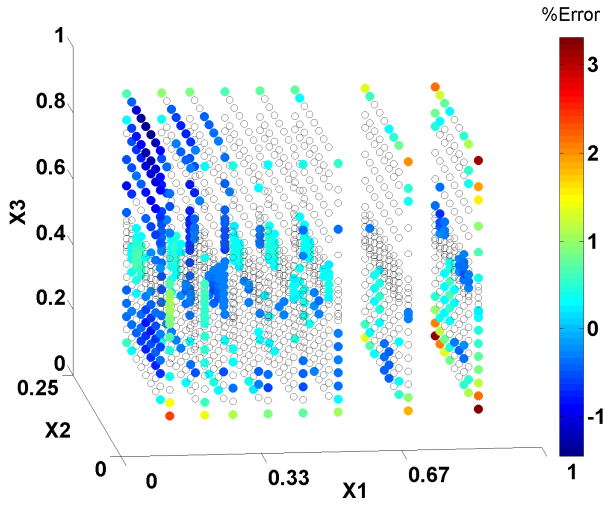

Figure 4 Data regions with RBF prediction error greater than 0.25 %.

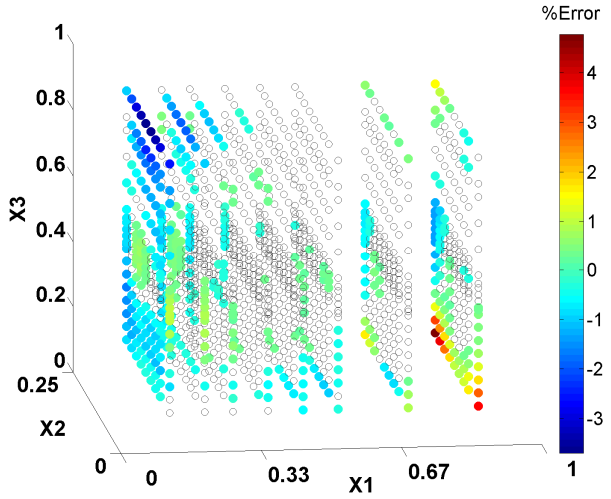

Figure 5 Regions with GP prediction error > 0.25 %.

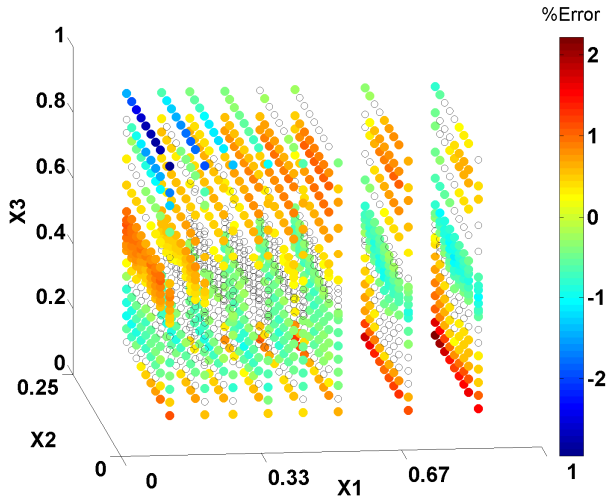

Figure 6 Data regions where non-linear regression predictions have error greater than 0.25 %.

The performance of ANN, GP and RBF models with 100 training points are very similar. However, ANN models were fastest to build followed by RBF models. GP model building was the most computationally costly. For example, on a dual core 2 GHz PC with 3 GB of RAM, the ANN models were built under a minute, RBF models were built in 3 minutes and GP model building took 6 minutes.

Clearly, the ANN model with 1000 training points has excellent performance with only 6 points having prediction errors greater than 0.25 %. The points with high error (>0.25%) lie on the boundaries of the dataset suggesting that the accuracy can be improved by extending the range of the data used to build models beyond the required limits.

The comparison between ANN with 1000 training points and GP, RBF models is not entirely fair because the ANN model used 1000 training points whereas the GP and RBF models only used 100. However, lower number of points was used for GP and RBF models due to their inefficiency in handling large datasets. Comparing RBF and GP models, GP performs better even though the maximum errors predicted by GP are higher than RBF predictions. This is because, GP predictions have very good accuracy in the central portion of the dataset compared to the RBF predictions with ~1% error in the same region (near 0.5,0.5,0.5). Also, the GP models perform poorly near the border of the data region. If we restrict the model predictions to near the central region, GP's performance would be comparable to ANN. This is a significant advantage in applications where evaluating each point is expensive and applications with detectable noise. If the signal to noise ratio in the dataset is low, one can use GP to build accurate models with filtered data.

The non-linear regression model performs poorly for this application. Even though the maximum number of training points was used to build the regression model, the prediction errors are high in the entire data space and not only at boundaries. The resulting regression model is the simplest of the models considered. However since the non-linearity of the dependent variable is not significant, one might expect the regression model to behave comparably to the RBF or GP models. Clearly, this is not the case. It should be noted that one could build more accurate regression models by transforming the data. This case is not shown here to preserve consistent processes for all the metamodels considered. Even with transformation, it may not be possible to reach the accuracy of ANN predictions.

Metamodel Smoothness at Off-grid Points

The above results only verify the accuracy requirement. The best metamodeling approach for this example should also satisfy the smoothness requirement. All the above metamodels were used to predict the dependent variable by fixing X2 and

X3 at a constant value and varying X1 from 0 to 1 in intervals of 4.5×10^{-5}. The results are shown in Figure 7. The real values are almost identical to the ANN model predictions. Only the N-LR model predicts the behavior similar to the actual data. Both RBF and GP models exhibit spurious oscillations.

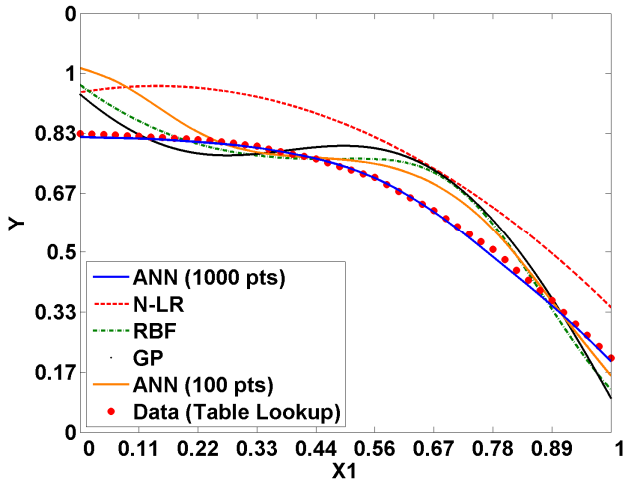

Figure 7 Smoothness of metamodel predictions with finely varying X1.

The results of varying X2 by 5.2×10^{-5} are shown in Figure 8. Although all metamodels predict varying degrees of nonlinear behavior, non of the models except ANN predict the true nonlinearity of the dataset.

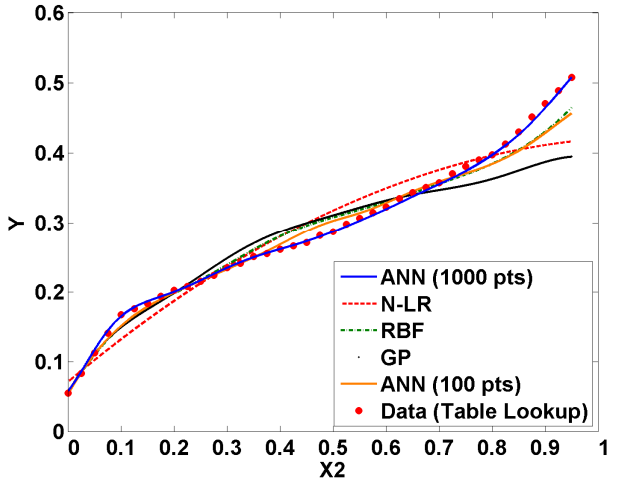

Figure 8 Checking smoothness of metamodels predictions by varying X2 in fine intervals.

A similar exercise was performed by fixing X1 and X2 at constant values and varying X3 in intervals of 1.8×10^{-4}. The predictions from all the above metamodels are shown in Figure 9. In this case, the GP model exhibits similar behavior to the actual data. The N-LR and RBF models do not even capture the correct behavior qualitatively.

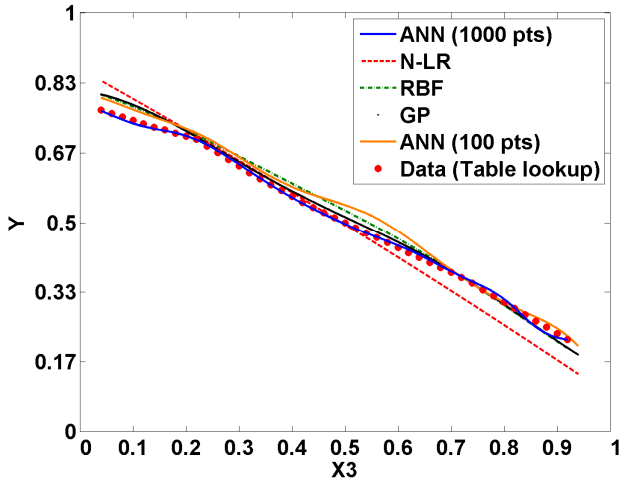

Figure 9 Finely varying X3 to capture metamodel inaccuracies.

Metamodel Interpolation Accuracy

To test the interpolation capabilities of the ANN model, we removed the data corresponding to X2 = 0.5 to build another set of ANN and regression models. These models were then used to predict at the removed points. This was done to calculate the interpolation error accurately.

The predictions from ANN and regression models are shown in Figure 10 and Figure 11 respectively. Clearly, the ANN model predictions satisfy both accuracy and smoothness requirements whereas the regression model predictions satisfy neither of the requirements.

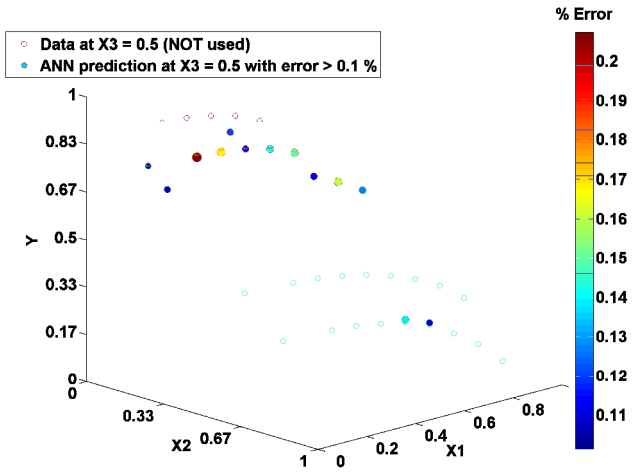

Figure 10 Interpolation error of ANN model at X3 = 0.5, colors correspond to errors.

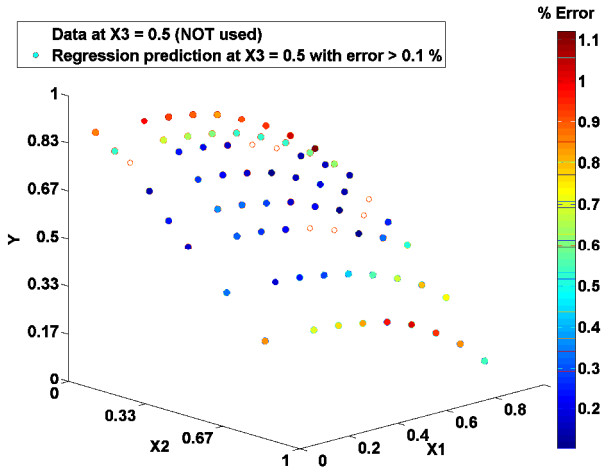

Figure 11 Interpolation error of regression model at X3 = 0.5, colors correspond to errors.

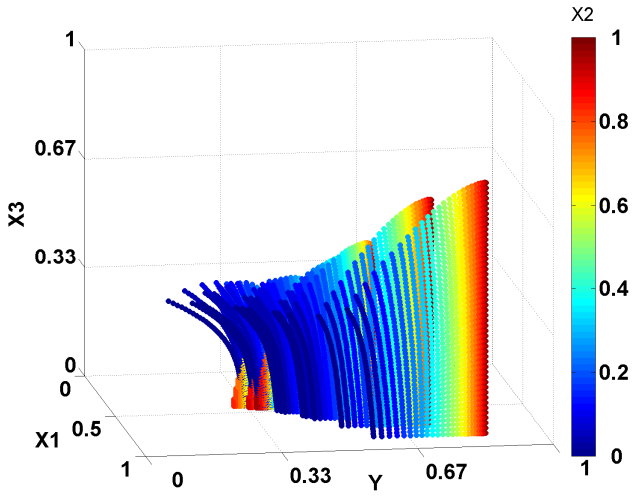

Figure 12 Representative dataset used in gas turbine applications.

Although 70 % of the points were used for training, the smoothness check verifies that the ANN model retained generality and accuracy. Thus, based on the accuracy and smoothness requirements, we can conclude that for the applications similar to the above example, ANN is the best choice.

APPLICATION -B

The data contains three independent variables and one dependent variable. The scaled dataset is shown in Figure 12. Figure 13 shows a 2 dimensional view of the data. The dataset is comprised of ~14000 points. It is representative of the nonlinearity and size of the datasets typically used in gas turbine applications. Clearly the non-linearity within the same family of curves and between separate families are significantly more pronounced than the previous example. The maximum allowable error is 1 % and the predictions are required to be smooth at non-training points. This application is a large dataset considering the strict accuracy limits required for metamodels predictions.

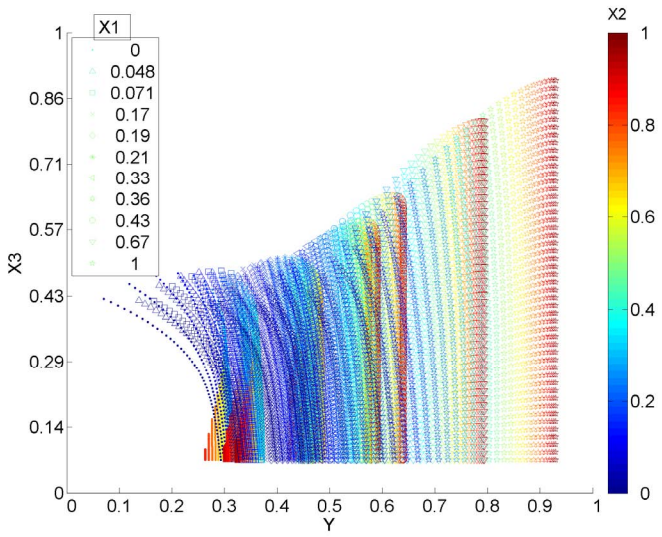

Figure 13 2D view of representative gas turbine data set.

A similar procedure as described in application A was used to build metamodels for this example. Non-linear regression models were built with 9750 points. RBF and GP models were built using 100 points closest to an OLH population. Two ANN models were built as before: one with 100 points used for GP and RBF models and another with 70 % of the points as training points. Thirty nodes were used in the sigmoid layer of the ANN model.

The accuracy of the various metamodel predictions are listed in Table 2. The mean (L1 norm), root mean-squared error (RMSE) and maximum absolute errors were calculated from predictions at all available data points.

Table 2 Prediction errors of various metamodels.

Model	Mean Error	RMSE	Max. Abs. Error (%)
ANN (100 pts)	2.72	5.97	20.2
RBF (100 pts)	1.79	3.71	23
GP (100 pts)	1.43	2.99	19.8
N-LR (9750 pts)	3.31	4.80	22.8
ANN (9750 pts)	0.15	0.23	0.89

From Table 2, it may seem that only the ANN model with large training points satisfies the accuracy requirements. However, focusing on regions with high error yields a better picture of model accuracy. The model predictions with error greater than 0.5 % are shown in Figures 14-17. The large errors in the ANN model (with 9750 training points) are limited to a few points at the corner of the data boundary as before. These points are not critical to the application and the error can be reduced significantly by increasing the data range beyond the operating limits of the metamodels.

Although the maximum errors predicted by RBF, GP and N-LR models are very high, the maximum error is only at a localized point on the corner of the dataset as seen in Figures 15-17.

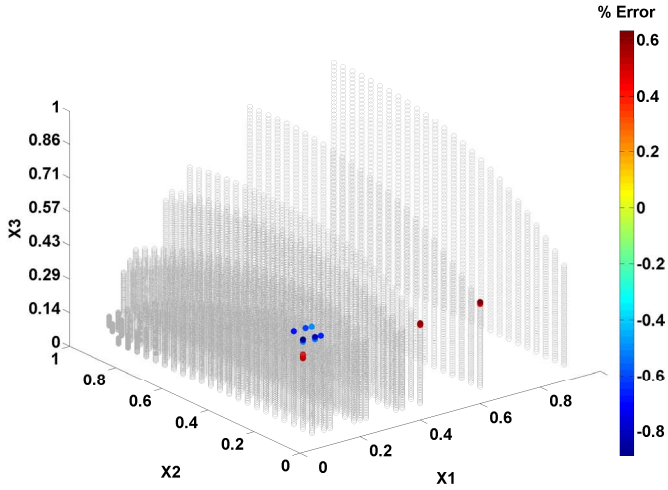

Figure 14 ANN (9750 training points) model predictions with error greater than 0.5 %.

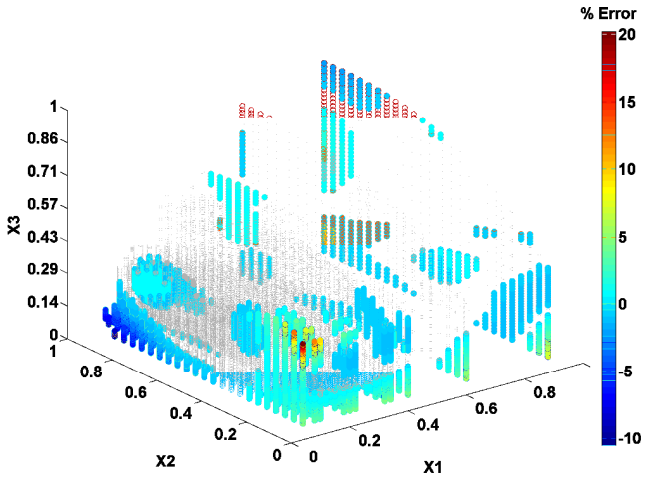

Figure 15 ANN (100 training points) prediction error.

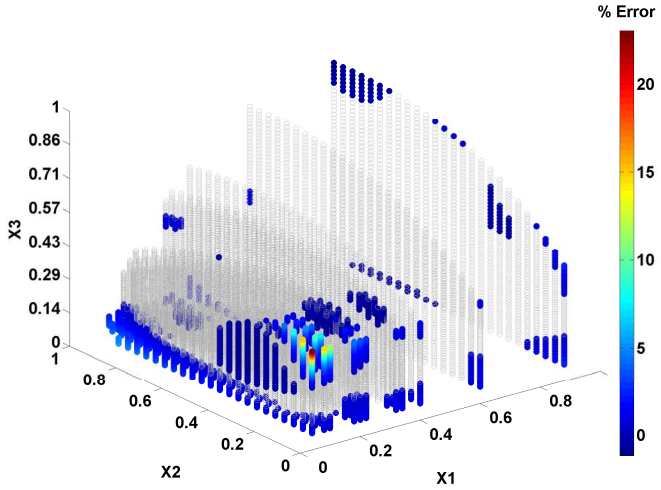

Figure 16 Data regions with RBF model prediction error greater than 0.5 %.

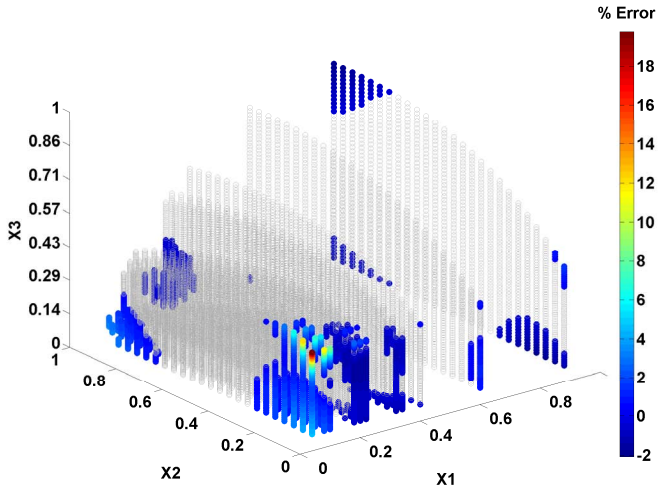

Figure 17 GP model prediction regions with error > 0.5%.

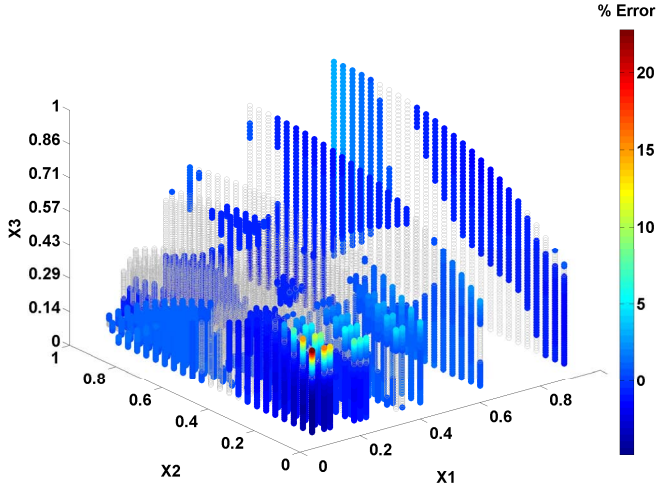

Figure 18 Regression model prediction regions with error greater than 0.5 %.

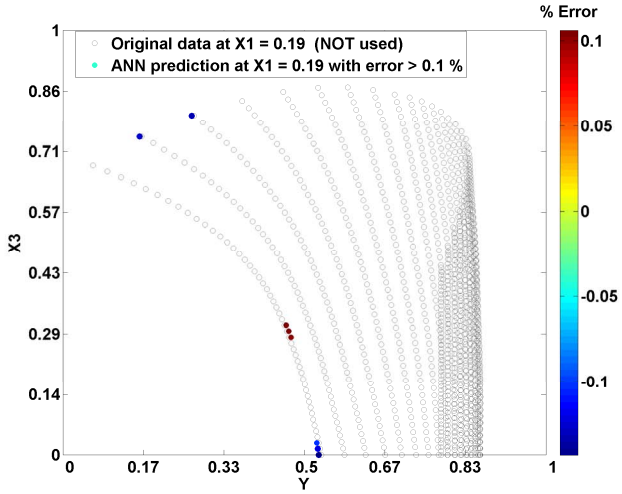

Figure 19 ANN predictions at the removed points with error greater than 0.1 %.

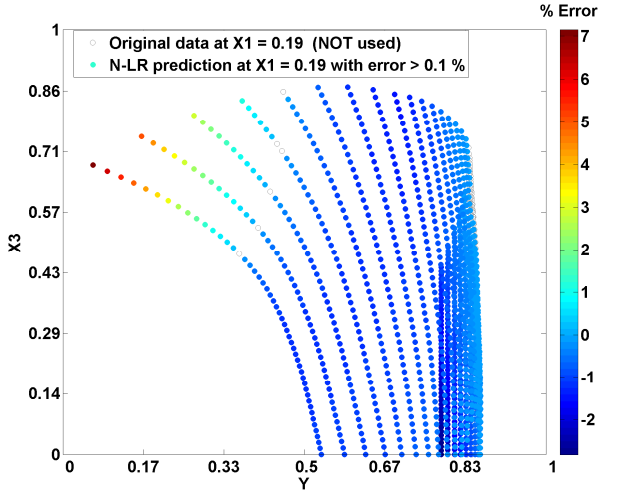

Figure 20 Non-linear regression predictions with errors greater than 0.1 %.

Metamodel Interpolation Accuracy

To test the interpolation capabilities of the ANN model, we removed the data corresponding to X1 = 0.19 to build another set of ANN models. These models were used to predict the dependent variable at the removed points. This was done to calculate the interpolation error accurately.

The predictions from ANN and N-LR models are shown in Figure 19 and Figure 20. The behavior is very similar to that observed in the previous application; ANN outperforms regression model in all cases.

The model build times for this application are listed in Table 3. The ANN model is the most efficient in terms of model building and model prediction time.

Table 3 Model build time comparison.

Model	Build Time
ANN (100 pts)	1-2 min
RBF (100 pts)	4-5 min
GP (100 pts)	5-10 min
N-LR (9750 pts)	1 min
ANN (9750 pts)	2-4 min

From the above results, ANN models clearly outperform the other three metamodeling techniques. Although for the same number of training points GP and RBF perform similarly to ANN models, the quantity of data is not a limiting factor for this application. Thus, when large data sets are available, GP and RBF do not provide any advantages over ANN models. In fact, due to the prohibitive optimization runs required by both GP and RBF models, it may not be possible to build highly accurate models when the data sets are large and the nonlinearities are severe.

CONCLUSIONS

A comparative study on accuracy and efficiency of metamodels constructed from large datasets was performed. Large datasets inspired from industrial applications were used to test several metamodeling techniques including Artificial Neural Networks, Radial Basis Functions, Gaussian Process and Non-linear regression. The examples are illustrative of a wide range of industrial applications in aircraft engines and gas turbines. Although Radial Basis Functions and Gaussian Process models are robust for small data sets, it was shown using multiple examples that they are not efficient enough when applied to large datasets with high nonlinearity.

It was found that when sufficiently large data is available, ANN models outperform the other three modeling techniques in terms of accuracy and prediction smoothness.

REFERENCE

[1] Wang, G. G. and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization", Journal of Mechanical Design, Vol. 129, No. 4., 2007, 370-380.

[2] Barthelemy, J. F. M. and Haftka, R., 1993, "Approximation Concepts for Optimal Structural Design – A Review", Structural Optimization, 5, 129-144.

[3] Simpson, T. W., Peplinski, J., Koch, P. N. and Allen, J. K., 2001, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations", Engineering with Computers, 17(2), 129-150.

[4] Jin, R., Chen, W. and Simpson, T. W., 2001, "Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria", Structural and Multidisciplinary Optimization, 23, 1-13.

[5] T. Hastie, R. Tibshirani and J. Friedman, 2003, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer Series in Statistics, Springer, New York.

[6] Cherkassy, V. and Mulier, F., 2007, "Learning from Data", 2nd Edition, Wiley, New York.

[7] Cover, T. and Thomas, J., 1991, "Elements of Information Theory", Wiley, New York.

[8] E. Acar, M. and Rais-Rohani, "Ensemble of metamodels with optimized weight factors", 2009, Structural and Multidisciplinary Optimization, 37, 279-294.

[9] S. Haykin, "Neural Networks: A comprehensive foundation", Prentice Hall, 1999.

[10] K. Mehrotra, C. K. Mohan and S. Ranka, "Elements of Artificial Neural Networks", 1996, MIT Press, Cambridge.

[11] C. M. Bishop, "Neural Networks for Pattern Recognition", 1996, Oxford University Press, USA.

[12] M. D. Buhmann, "Radial Basis Functions: Theory and Implementations", 2003, Cambridge Monographs on Applied and Computational Mathematics, 12, Cambridge University Press, Cambridge.

[13] R. J. Howlett and L. C. Jain, "Radial Basis Function Networks 1: Recent Developments in Theory and Applications", 2001, Physica-Verlag Heidelberg.

[14] C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Machine Learning", 2006, MIT Press, Cambridge.

[15] J. Q. Shi, "Gaussian Process Regression Analysis for Functional Data", 2011, CRC Press.

[16] A. R. Gallant, "Nonlinear Statistical Models", Wiley Series in Probability and Statistics, John Wiley & Sons, New York.

[17] D.M. Bates and D. G. Watts, "Nonlinear Regression Analysis and Its Applications", 1988, Wiley Series in Probability and Statistics, John Wiley & Sons, New York.

[18] R. K. Beatson, J.B. Cherrie and C.T. Mouat, "Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration", 1999, Advances in Computational Mathematics, 11, 253-270.

[19] R. L. Hardy, "Theory and applications of the multiquadratic-biharmonic method", 1990, Computational Methematics and Applications, 19, 163-208.

[20] R.K. Beatson and W.A. Light, "Fast evaluation of radial basis functions: I", Computational Mathematics and Applications, 24, 1992, 7-19.

[21] R.K. Beatson and W.A. Light, "Fast evaluation of radial basis functions: Methods for 2-dimensional polyharmonic splines", Journal of Numerical Analysis, 1997, 17, 343-372.

[22] R. P. Lippmann, "An introduction to computing with neural nets", 1987, IEEE, ASSP Magazine, Apr, 1987, 4-22.

[23] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. “Learning internal representations by error propagation”, 1986, 318-362, MIT Press, Cambridge, MA.
[24] K. Levenberg, “A method for the solution of certain non-linear problems in least squares”, 1944, The Quarterly of Applied Mathematics, 2, 164-168.
[25] Gill, P. R., Murray, W., and Wright, M. H., “The Levenberg-Marquardt Method”, 1981, Practical Optimization, Academic Press, London, 136-137.
[26] Marquardt, D. “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”, SIAM Journal of Applied Mathematics, 1963, 11, 431-441.